3DCNN Performance in Hand Gesture Recognition Applied to Robot Arm Interaction

J. A. Castro-Vargas[1], B. S. Zapata-Impata[1], P. Gil[1,3], J. Garcia-Rodriguez[2,3] and F. Torres[1,3]
[1]*Dept. of Physics, Systems Engineering and Signal Theory, University of Alicante, San Vicente del Raspeig, Alicante, Spain*
[2]*Dept. of Computer Technology, University of Alicante, San Vicente del Raspeig, Alicante, Spain*
[3]*Computer Science Research Institute, University of Alicante, San Vicente del Raspeig, Alicante, Spain*
{*jacastro, brayan.impata, pablo.gil, jgr, fernando.torres*}*@ua.es*

Keywords: Gesture Recognition from Video, 3D Convolutional Neural Network.

Abstract: In the past, methods for hand sign recognition have been successfully tested in Human Robot Interaction (HRI) using traditional methodologies based on static image features and machine learning. However, the recognition of gestures in video sequences is a problem still open, because current detection methods achieve low scores when the background is undefined or in unstructured scenarios. Deep learning techniques are being applied to approach a solution for this problem in recent years. In this paper, we present a study in which we analyse the performance of a 3DCNN architecture for hand gesture recognition in an unstructured scenario. The system yields a score of 73% in both accuracy and F_1. The aim of the work is the implementation of a system for commanding robots with gestures recorded by video in real scenarios.

1 INTRODUCTION

Interaction with robots is often carried out in industrial environments with control panels. However, control panels are not often user-friendly. Ideally, in service tasks, their use for interacting with people should be intuitive and user-friendly through the use of voice and gestures. The use of voice is usually constrained by the environmental noise. Therefore, visual methods are interesting for human-robot interaction, specifically, in the recognition of hand gestures to indicate actions to the robot.

In most of the works in human-robot interaction, hand gestures are performed at close ranges from the camera in order to capture more details (Luo and Wu, 2012), (Malima et al., 2006) and (Ionescu et al., 2005), but sometimes this can be dangerous if the user has to invade the workspace of the robot to reach that distance with respect to the camera. Other approaches are constrained to work in structured scenarios, i.e. scenes with a previously defined background or easily segmentable as in (Chen et al., 2010). As a result, the classification of gestures for operating robots is more reliable but requires modifying the environment.

Besides, it is possible to find methods oriented to the detection of static gestures by pose as in (Luo and Wu, 2012) and (Malima et al., 2006) and dynamic gestures carried out in a temporal band as in (Ionescu et al., 2005), (Abid et al., 2014) and (Strobel et al., 2002). However, pose-based methods limit the movements that the user can perform in front of the robot, because some gesture could trigger the system and result in a high number of false positives. For this reason, we propose the use of temporal hand gestures, in order to have more security when interacting with the robot, minimizing the risk of false positives.

Recently, deep learning techniques as shown in (Schmidhuber, 2015) are being applied to find a solution for this problem. In this line, methods based on Convolutional Neural Networks (CNN) trained with sequences are interesting because of their robustness at learning visual features. Previously, CNN (Barros et al., 2014) and ConvLSTM (Tsironi et al., 2016) have been used in robotic applications.

The main contributions of this work are two folded. First, we propose and evaluate the use of a 3DCNN (Molchanov et al., 2015) to classify hand gestures. Secondly, in order to evaluate its performance, we have generated a novel dataset[1] of four complex gestures performed at a distance between 1.5 and 2.5m from a RGB-D camera in an unstrctured environment.

The document is organised as follows: Section 2

[1]https://drive.google.com/drive/folders/1-BTbcZ6vXLDQ6jj5ptC3SF2cMkBAOsfA?usp=sharing

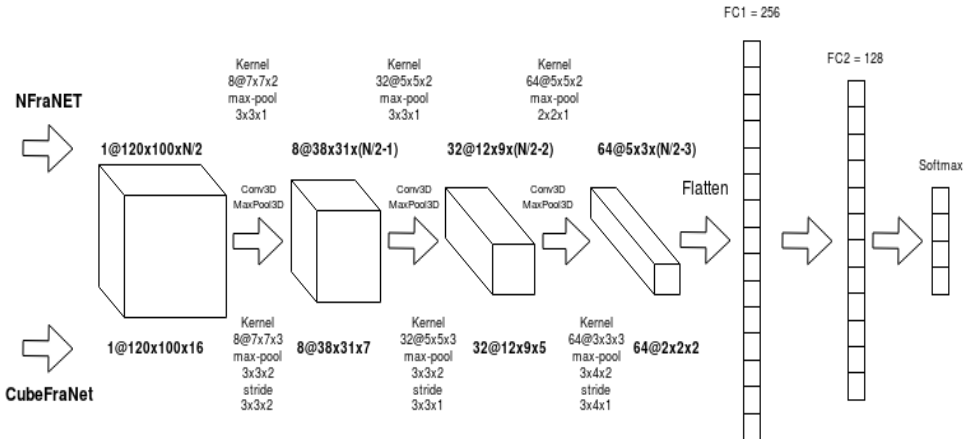

Figure 1: Architecture implemented in this work. Top pipeline (NFraNet) and bottom pipeline (CubeFraNet) represent the two configurations tested.

presents the evaluated neural network and describes the dataset used in the experimentation. Section 3 describes the proposed learning methodology and the evaluation metrics used to measure the performance of the recognition system. Section 4 details the experiments and discusses the obtained results. Finally, the conclusions of this work are presented in section 5.

2 NETWORK ARCHITECTURE AND DATASET

The 3DCNN architecture implemented in this work consists of three 3D convolutional layers, each layer followed by a max pooling layer. This way, the 3D layer is partitioned and replaced with a downsampling layer by the activation of its maximally active unit, as is described in (Giusti et al., 2013). The output of the last convolutional layer is used as input for two consecutive fully connected layers and then one softmax layer (Figure 1).

The weights of each convolutional layer have been initialized according to (Molchanov et al., 2015), from a uniform distribution defined between $[-W_b, W_b]$ where $W_b = \sqrt{6/(n_i + n_o)}$, n_i are the input neurons and n_o the output neurons.

The dataset[1] contains 624 sequences of RGB-D images captured from a D435 depth camera, manufactured by Intel RealSense, at a distance of 1.5 and 2.5m with two fellows, one male and one female. The distance was defined taking into account the working environment of the robot. The images were stored with a resolution of 640 x 360 pixels.

We changed the two people clothing every 104 samples of the 624 sequences/scenes. Thus, 6 different outfits were used. In addition, the dataset was recorded balancing the classes: there are 104 sequences for each of the 4 gestures. Finally, in each sequence, the recorded gesture was repeated twice. The figure 2 shows a subset of the gestures.

Figure 2: Example from our dataset of each gesture classified in this work, with different distances (from top to bottom, left to right): down, up, right and left movements.

3 LEARNING METHODOLOGY

Initially, we performed a brief evaluation to select an optimal number of frames from the video sequences, experimenting with selecting 4, 8, 16 and 32 frames (Figure 3). We computed the accuracy rate of the proposed architecture for each test, being 16 the best number of frames to be used according to the obtained results (Figure 3).

Besides, for this work, we use a minimum max-pooling on the depth to adapt the network to the frames chosen. Later, we modify the kernels and max-pooling of the network to work with the size of the selected frames. Additionally, the training samples are pre-processed to facilitate the learning of the

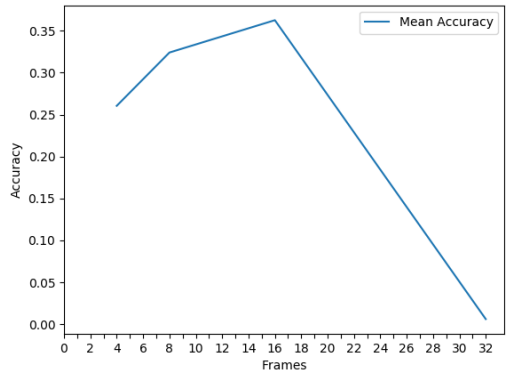

Figure 3: Analysis of accuracy using 4, 8, 16 and 32 frames with an L2 regularization.

network. We augment the dataset before the training phase and the images are resized to 320 x 180 pixels.

Afterwards, depending on the number of frames that we want to use for training, we select N equidistant frames over the total length of the sequence (Figure 4 illustrates this process). Then, other additional M sets are selected by taking the next frame of the sequence, obtaining M+1 sets of N frames per scene. Moreover, since each sequence holds the same gesture repeated twice, each set of N frames is split in half to create a group of N/2 frames with the first repetition and another group of N/2 frames with the second repetition.

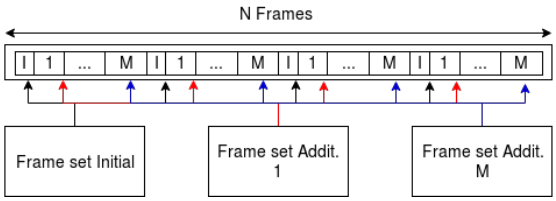

Figure 4: Selection of frames throughout the pre-processing of scenes.

In Table 1 can be observed the result of applying the selection of frames over the dataset. Depending on the number of frames selected for training, the total number of samples is reduced and the spatiotemporal information in each sample varies. For example, with a selection of N=8 frames, after splitting them in half and selecting the M additional sets, we have a total of 7488 samples of 4 frames.

In order to find that selecting 16 frames yielded a greater performance, we trained the NFraNet network, defined in Figure 1, with the samples split as shown in Table 1. This experiment was performed running 600 epochs on mini batches with a size of 16 samples. Concurrently, while a batch was running on a Nvidia GeForce GTX 1080ti GPU, the following batch received an extra online pre-processing: images were rotated with a 50% probability between $[-3,3]$ degrees. Then, a crop was made on the centre with and offset of -10 pixels in width and -15 pixels in height. This crop had a 50% of probability to undergo random displacements between $[-2,2]$ pixels in height and $[-4,4]$ pixels in width. This preprocessing was carried out with the goal of providing the network with increased online data and improving its generalization capabilities. In order to avoid overfitting the training set, two regularization techniques were used during training. We applied Dropout (Srivastava et al., 2014) to the units of the network with a 50% of probability and weights were regularized using L_2 regularization (Goodfellow et al., 2016), with beta = 0.01.

Table 1: Number of samples used for training after applying both sampling and selection of additional frames. *Addit.* stands for the number of additional frame sets.

Frames (N)	Addit. (M)	Samples	S. Split
8	5	3744	7488
16	4	3120	6240
32	1	1248	2496
64	0	624	1248

4 EXPERIMENTS AND RESULTS

We have studied the performance of the NFraNet and the CubeFraNet during training by applying k-fold cross-validation (Kohavi, 1995) with k=5 for both of them. The performance evaluation of our proposal has been carried out with the following metrics (Powers, 2011), widely used for this kind of tasks: Accuracy (*Acc*), Precision (*P*), Recall (*R*) and F1-measure (F_1). They are defined as follows:

$$\begin{aligned} Acc &= (TP+TN)/(TP+FP+FN+TN) \\ P &= TP/(TP+FP) \\ R &= TP/(TP+FN) \\ F_1 &= 2*(P*R)/(P+R) \end{aligned} \quad (1)$$

where TP (True Positives) denote the numbers of correctly detected instances like the correct gesture; FN (False Negatives) denote the number of instances that, although having performed the gesture, it has not been detected as such; FP (False Positives) indicate the number of times that a certain gesture has been recognized as the wanted one, despite having executed another; TN (True Negatives) indicates the number of times that the executed gesture has been correctly identified as not belonging to the wanted one.

A cross-validation experiment was made to compare both architectures but they showed a similar performance. Figure 5 shows the accuracy rate in cross validation of one of them (NFraNet). Since the networks showed to suffer from overfitting, we decided to measure their final performance using a test set after tuning them using cross-validation on the training set.

The test set consisted of 20 unknown scenes samples. These samples were never seen in training, so with them we evaluated the generalization capability of the proposed neural network. In this evaluation, we obtained a 0.728 success rate for the first configuration of the 3DCNN network (NFraNet) and a 0.684 success rate for the second (CubeFraNet) in terms of accuracy.

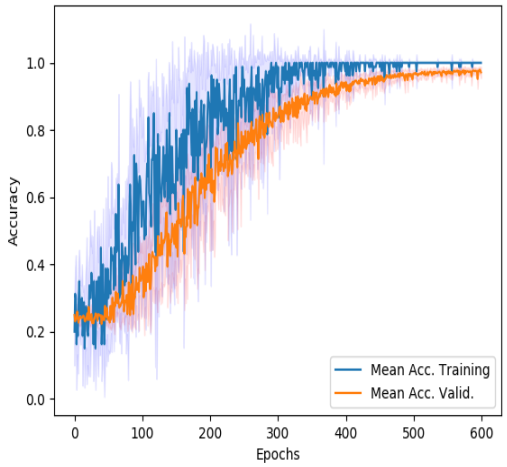

Figure 5: Average accuracy obtained with 5-fold cross-validation with NFraNet.

Table 2 holds the scores achieved by NFraNet for the recognition task of the four gestures described in Section 2 on the test set. The table shows that 'Right' obtained the best score in F_1 and P (0.85 and 0.78, respectively) and the second best in Acc and therefore, it is the most reliable prediction of the four gestures if we would tele-operate a robot. 'Up' is the best one considering Acc and R (0.81 in both cases) and the second best valued gesture by F_1, however its P is the lowest. 'Left' and 'Down' have similar values in both Acc and R (around 0.66). However, between both gestures, 'Left' predictions are more reliable as indicated by the P.

A more exhaustive study can be carried out observing the confusion matrix in figure 6. It indicates a tendency to confuse 'Left' and 'Up' gestures with 'Down', while 'Right' shows a certain level of robustness to be identified, as indicated by its scores. Besides, the network tends to predict in favor of 'Up', affecting even the 'Right' gesture, which seems to differ more from the rest of the gestures.

Table 2: The Accuracy, Precision, Recall and F1 score achieved by NFraNet on the test set.

Gesture	Acc.	P	R	F_1
Down	0.663	0.654	0.663	0.658
Left	0.650	0.754	0.650	0.698
Right	0.788	0.940	0.788	0.857
Up	0.813	0.631	0.813	0.710
Average	0.728	0.745	0.728	0.731
Std	0.073	0.122	0.073	0.075

In relation to the issue of gestures being confused with 'Down', we think that this could be due to the body posture when the gestures were recorded. As can be seen in the Figure 2, during the execution of the gestures 'Down', 'Left' and 'Up' the elbow is placed almost identically. However, in order to perform the 'Right' gesture, it elbow is raised to a different position. Owing to this, the network could have given more importance to spatial features related to the arm position than to temporal features related to the hand.

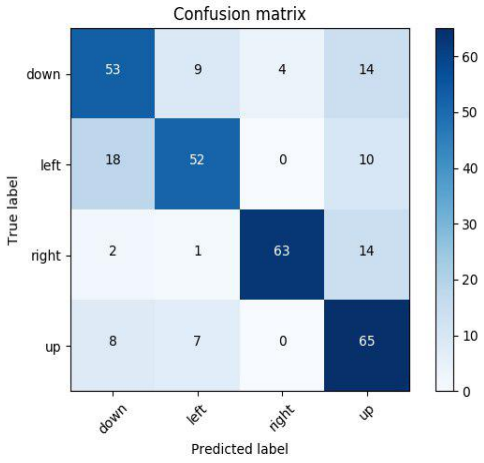

Figure 6: Confusion matrix of the NFraNet on the test set.

5 CONCLUSIONS

In this work, two neural network configurations based on 3DCNN have been tested in order to classify 4 gestures for tele-operating or interacting with robots at a safe distance. The best proposed configuration (NFraNet) yielded an average accuracy of 73% and the resulting confusion matrix showed us that it effectively extracted spatiotemporal features with which we could classify the gestures with reliability, compared to the state of the art. Taking into account

the number of samples and the visual and temporal complexity of the generated novel dataset, the performance of the presented system is close to that of (Molchanov et al., 2015), in which they achieved 77.5% using a visually segmented dataset for the recognition of hand gestures.

Taking into account the present results, the proposed method could stand as a starting point for a future work in the field of human-robot interaction with gestures. In a real world scenario, an adaptation phase of the user to the constrains of our system would be required. Also, a high level configuration of the neural network could be carried out to increase the confidence of the outputs. As a benefit, during online execution, the system would have several opportunities to identify the same gesture as the temporal window moves.

In the future, we plan to test the possibility of using an already trained network in a similar visual task and transfer its features to our problem. In addition, checking the performance of a mixed architecture using ConvLSMT and 3DCNN would be an interesting experiment. Finally, in order to generate more samples we would like to experiment with methods that produce synthetic data realistic enough for learning.

ACKNOWLEDGEMENTS

This work was funded by the Ministry of Economy, Industry and Competitiveness from the Spanish Government through the DPI2015-68087-R and the predoctoral grant BES-2016-078290, by the European Commission and FEDER funds through the project COMMANDIA (SOE2/P1/F0638), action supported by Interreg-V Sudoe.

REFERENCES

Abid, M. R., Meszaros, P. E., Silva, R. F., and Petriu, E. M. (2014). Dynamic hand gesture recognition for human-robot and inter-robot communication. In *IEEE Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, pages 12–17.

Barros, P., Parisi, G. I., Jirak, D., and Wermter, S. (2014). Real-time Gesture Recognition Using a Humanoid Robot with a Deep Neural Architecture.

Chen, K.-Y., Chien, C.-C., Chang, W.-L., and Teng, J.-T. (2010). An integrated color and hand gesture recognition approach for an autonomous mobile robot. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 5, pages 2496–2500. IEEE.

Giusti, A., Cireşan, D. C., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2013). Fast image scanning with deep max-pooling convolutional neural networks. In *International Conference on Image Processing (ICIP)*, pages 4034–4038. IEEE.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Ionescu, B., Coquin, D., Lambert, P., and Buzuloiu, V. (2005). Dynamic hand gesture recognition using the skeleton of the hand. *Eurasip Journal on Applied Signal Processing*, 2005(13):2101–2109.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1137–1143.

Luo, R. C. and Wu, Y. C. (2012). Hand gesture recognition for Human-Robot Interaction for service robot. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 318–323. IEEE.

Malima, A., Özgür, E., and Çetin, M. (2006). A fast algorithm for vision-based hand gesture recognition for robot control. In *IEEE 14th Signal Processing and Communications Applications Conference*, pages 6–9. IEEE.

Molchanov, P., Gupta, S., Kim, K., Kautz, J., and Clara, S. (2015). Hand Gesture Recognition with 3D Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–7.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014(15):1929–1958.

Strobel, M., Illmann, J., Kluge, B., and Marrone, F. (2002). Using spatial context knowledge in gesture recognition for commanding a domestic service robot. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 468–473.

Tsironi, E., Barros, P., and Wermter, S. (2016). Gesture recognition with a convolutional long short-term memory recurrent neural network. *Bruges, Belgium*, 2.